Nicht für den Prüfling bestimmt

I Erläuterungen

Voraussetzungen gemäß KCBG und Abiturerlassen BG jeweils in der für den Abiturjahrgang geltenden Fassung

Standardbezug
Die nachfolgend ausgewiesenen Kompetenzbereiche sind für die Bearbeitung der jeweiligen Aufgabe besonders bedeutsam. Darüber hinaus können weitere, hier nicht ausgewiesene Kompetenzbereiche für die Bearbeitung der Aufgabe nachrangig bedeutsam sein, zumal die Kompetenzbereiche in engem Bezug zueinanderstehen. Die Operationalisierung des Bezugs zu den Kompetenzbereichen des Standardbezugs erfolgt in Abschnitt II.

Aufgabe	Kompetenzbereiche				
	K1	K2	K3	K4	K5
1.1		X			
1.2	X				
1.3		X	X		
1.4.1				X	
1.4.2		X			
1.5.1	X		X		
1.5.2				X	
1.6			X		
2.1	X	X			
2.2		X			
2.3		X			X
2.4.1				X	
2.4.2				X	
2.4.3				X	
2.4.4				X	
2.5	X		X		

Inhaltlicher Bezug
Die nachfolgend ausgewiesenen Themenfelder sind die wesentliche inhaltliche Grundlage für die vorliegenden Aufgaben. Darüber hinaus können weitere, hier nicht explizit ausgewiesene Themenfelder für die Bearbeitung nachrangig bedeutsam sein.
Q1: Objektorientierte Softwareentwicklung
Q2: Datenbanksysteme
Q3: Datenkommunikation
verbindliche Themenfelder: Objektorientierte Modellierung (Q1.1), Implementierung von Klassen und Assoziationen (Q1.2), Suchen und Sortieren (Q1.3), Konzeptionelle und logische Modellierung einer Datenbank (Q2.1), Datenabfrage und Datenmanipulation mit SQL (Q2.2), Serielle Schnittstelle (Q3.1)

II Lösungshinweise

In den nachfolgenden Lösungshinweisen sind alle wesentlichen Gesichtspunkte, die bei der Bearbeitung der einzelnen Aufgaben zu berücksichtigen sind, konkret genannt und diejenigen Lösungswege aufgezeigt, welche die Prüflinge erfahrungsgemäß einschlagen werden. Selbstverständlich sind jedoch Lösungswege, die von den vorgegebenen abweichen, aber als gleichwertig betrachtet werden können, ebenso zu akzeptieren.

Nicht für den Prüfling bestimmt

Aufg.	erwartete Leistungen	BE I	BE II	BE III
1.1	beschreiben Das Klassendiagramm besteht aus den Klassen `MAPD`, `Mitarbeiter`, `Probe`, `Hersteller`, `Messwertsatz`, `Messwert` und `Messstation`. Eine Klasse beschreibt dabei gleichartige Objekte mit ihren Eigenschaften (Attribute) und ihrem Verhalten (Methoden). Im gegebenen Modell besitzt beispielsweise die Klasse `Messwertsatz` die Attribute `eigenschaft`, `einheit`, `min` und `max` sowie die Methoden `hinzuguegenWerte()` und `ermittleMedian()`. Eine Klasse ist vergleichbar mit einem Bauplan für Objekte. Die zwischen den einzelnen Klassen bestehenden Assoziationen werden durch Linien zwischen den daran beteiligten Klassen dargestellt. Die Multiplizität bezeichnet die Wertigkeit einer Assoziation, d.h. sie spezifiziert die Anzahl der an der Assoziation beteiligten Objekte (hier: * für „0, 1 oder mehrere" und 1 für „genau ein"). Die Klasse `MAPD` ist die Hauptklasse des Systems, sie kennt die Proben, Hersteller, Mitarbeiter und Messstationen über unidirektionale Assoziationen. Das bedeutet, dass Proben-, Hersteller-, Mitarbeiter- und Messstationenobjekte das MAPD-Objekt nicht kennen. In diesen Beziehungen treten beispielsweise `Mitarbeiter` in der Rolle `mitarbeitende` auf. Die Multiplizität der Beziehung besagt, das MAPD-Objekt kennt 0, 1 oder mehrere `mitarbeitende`.	5		
1.2	zeichnen : Probe kennung = PAP14783 bezeichnung = Universalpappe Robust farbe = grau bauform = c-Welle : Hersteller name = Papierwerk strasse = Papierwerkstraße 1 ort = 63777 Aschaffenburg : Messwertsatz eigenschaft = Gewicht einheit = g/m² min = 295.3 max = 304.7 : Mitarbeiter nachname = Beate vorname = Salari kuerzel = BS : Messwertsatz eigenschaft = Berstdruck einheit = mpa min = 100.5 max = 240.5 : Messwert wert = 295.3 messzeit = 14.05.2023 10:00:05 : Messwert wert = 304.7 messzeit = 14.05.2023 11:00:05 : Messwert wert = 300.0 messzeit = 14.05.2023: 12:00:05 : Messwert wert = 240.5 messzeit = 14.05.2023 10:10:04 : Messwert wert = 100.5 messzeit = 14.05.2023 12:09:45	7		

1.3	überführen, implementieren
	`public class Probe {`
	`  private String kennung;`
	`  private Hersteller hersteller;`
	`  private String bezeichnung;`
	`  private String farbe;`
	`  private String bauform;`
	`  private DateTime eingangsdatum;`
	
	`  private List<Messwertsatz> mws;`			

Nicht für den Prüfling bestimmt

Aufg.	erwartete Leistungen	BE I	BE II	BE III
	(Code siehe unten)			

```
  public Probe(String kennung, Hersteller h, String
      bezeichnung, String farbe, String bauform, DateTime
      eingangsdatum) {
    this.kennung = kennung;
    this.hersteller = h;
    this.bezeichnung = bezeichnung;
    this.farbe = farbe;
    this.bauform = bauform;
    this.eingangsdatum = eingangsdatum;

    mws = new List<>();
  }


  public void setzeMitarbeiter(Mitarbeiter mitarbeiter,
                  String eigenschaft, String einheit){
    boolean gefunden = false;
    for (Messwertsatz mwsatz : mws) {
      if (mwsatz.getEigenschaft().equals(eigenschaft)) {
        mwsatz.setMitarbeiter(mitarbeiter);
        gefunden = true;
      }
    }
    if (!gefunden) {
      Messwertsatz mw = new Messwertsatz(eigenschaft,
                              einheit, mitarbeiter);
      mws.add(mw);
    }
  }


  public String erstelleProtokoll() {
    String ausgabe = "";
    ausgabe = ausgabe + "Kennung: " + this.kennung + "\n";
    ausgabe = ausgabe + "Hersteller: " +
    this.hersteller.getName() + "\n";
    ausgabe = ausgabe + "Bezeichnung: " + this.bezeichnung
    + "\n";
    for (Messwertsatz mwSatz : mws) {
      ausgabe = ausgabe + "Eigenschaft: " +
                  mwSatz.getEigenschaft() + "\n";
      ausgabe = ausgabe + "Einheit: " +
mwSatz.getEinheit()           + "|";
      ausgabe = ausgabe + "Min: " + mwSatz.getMin() + "|";
      ausgabe = ausgabe + "Max: " + mwSatz.getMax();
    }
    return ausgabe;
  }
```

<table>
<tr><th rowspan="2">Aufg.</th><th rowspan="2">erwartete Leistungen</th><th colspan="3">BE</th></tr>
<tr><th>I</th><th>II</th><th>III</th></tr>
<tr><td></td><td>

```
  public void erfasseMesswert(double mw,
                DateTime messzeit, String eigenschaft) {
    for (Messwertsatz mwSatz : mws) {
      if (mwSatz.getEigenschaft().equals(eigenschaft)) {
        mwSatz.hinzufuegenWert(mw, messzeit);
      }
    }
  }
}
```

überführen
implementieren</td><td>3</td><td>4</td><td>1</td></tr>
<tr><td>1.4.1</td><td>implementieren

```
public double ermittleMedian() {
  double median;
  Messwert temp;
  boolean sortiert = false;
  while (!sortiert) {
    sortiert = true;
    for (int i = 0; i < messwerte.size() - 1; i++) {
      if (messwerte.get(i).getWert() >
                messwerte.get(i + 1).getWert()) {
        temp = messwerte.get(i);
        messwerte.add(i, messwerte.get(i + 1));
        messwerte.add(i + 1, temp);
        sortiert = false;
      }
    }
  }
  if (messwerte.size() % 2 != 0) {
    median = messwerte.get(
                (messwerte.size() / 2)).getWert();
  }else{
    median = (messwerte.get(
              messwerte.size()/2).getWert() +
              messwerte.get(
              messwerte.size()/2-1).getWert())/2.0;
  }
  return median;
}
```

</td><td></td><td>4</td><td>3</td></tr>
<tr><td>1.4.2</td><td>implementieren

```
public void hinzufuegenWert(double mw, DateTime messzeit){
  int pos = 0;

  while(pos < messwerte.size()){
    if (messwerte.get(pos).getWert() > mw) {
      break;
    }
    pos++;
  }
```

</td><td></td><td></td><td></td></tr>
</table>

<table>
<tr><th rowspan="2">Aufg.</th><th rowspan="2">erwartete Leistungen</th><th colspan="3">BE</th></tr>
<tr><th>I</th><th>II</th><th>III</th></tr>
<tr><td></td><td><pre>
  messwerte.add(pos, new Messwert(messzeit, mw));

  this.min = messwerte.get(0).getWert();
  this.max = messwerte.get(messwerte.size()-1).getWert();
}
</pre>beschreiben
In der Methode <code>ermittleMedian()</code> muss die Logik zum Sortieren der Messwerte entfernt werden, da die Werte schon in sortierter Reihenfolge vorliegen. Der Teil zur Ermittlung des Medians muss weiterhin bestehen bleiben.</td><td></td><td>3

2</td><td>4</td></tr>
<tr><td>1.5.1</td><td>entwickeln, zeichnen

holeKraft():double
antwort:= NAK
kraft:= 0
messwerte:= [ ]
com.open() ? (T / F)
T: von i:=0 bis i<5, Schrittweite 1
  messwert:= com.read()
  i < 4 ? (T / F)
    T: messwerte[i]:= messwert
    F: messwert = EOT ? (T / F)
      T: antwort:=ACK
      F: ∅
F: ∅
com.write(antwort)
kraft:= konvertiereMesswert(messwerte)
Rückgabe: kraft

entwickeln
zeichnen</td><td>

2</td><td>4</td><td></td></tr>
</table>

Nicht für den Prüfling bestimmt

<table>
<tr><th rowspan="2">Aufg.</th><th rowspan="2">erwartete Leistungen</th><th colspan="3">BE</th></tr>
<tr><th>I</th><th>II</th><th>III</th></tr>
<tr><td>1.5.2</td><td>implementieren
<pre>
  public void erwarteMessung(Probe probe){
  if(this.messArt == 0){
    String resulttext = "";
    byte wert = com.read();
    while(wert != EOT){
      resulttext = resulttext + (char)wert;
      wert = com.read();
    }
    com.write(ACK);
    String[] r = resulttext.split(" ");
    double mw = Double.parseDouble(r[0]);
    probe.erfasseMesswert(mw, new DateTime(), eigenschaft);
  }else if(this.messArt == 1){
    double kraft;
    double kraftMax = 0.0;
    do{
      kraft = holeKraft();
      if (kraft > kraftMax) {
        kraftMax = kraft;
      }
    }while(kraft > (kraftMax * 0.9));
      probe.erfasseMesswert(kraftMax, new DateTime(),
                                   eigenschaft);
  }
}
</pre></td><td></td><td>3</td><td>5</td></tr>
</table>

Aufg.	erwartete Leistungen	BE I	BE II	BE III
1.6	entwickeln, zeichnen			
	entwickeln		4	4
	zeichnen	2		
	Summe 60	**19**	**24**	**17**

Aufg.	**erwartete Leistungen**	**BE**		
		I	**II**	**III**
2.1	beschreiben Ein Entity-Relationship-Diagramm besteht aus Entitätstypen und Relationen. Die Entitätstypen in Material 5 lauten Hersteller, Mitarbeiter, Charge, Produkt, Eigenschaft und Prüfgerät und werden als Rechtecke dargestellt. Ein Entitätstyp hat Attribute, die in Ellipsen/Kreisen dargestellt werden. Im angegebenen Diagramm sind z. B. die Attribute des Entitätstypen Eigenschaft eid, einheit und bezeichnung. Unterstrichene Attribute sind Primärschlüssel. Ein Primärschlüssel identifiziert eine Entität eindeutig innerhalb eines Entitätstypen. Zwischen Entitätstypen können Beziehungen existieren, diese werden mithilfe einer Raute dargestellt und benannt. Im gegebenen Diagramm existiert z. B. die Beziehung Mitarbeiter benutzt Prüfgerät. Die Kardinalität gibt das Mengenverhältnis einer Relation an. Im angegebenen Diagramm beispielsweise: Ein Mitarbeiter benutzt ein oder mehrere Prüfgeräte. Ein Prüfgerät wird von einem oder mehreren Mitarbeitern benutzt. Die Kardinalitäten wurden hier in der [min, max]-Notation angegeben, wobei der erste Wert der Minimalwert ist und der zweite Wert der Maximalwert.	5		

2.2	überführen
	`Hersteller(hid, firmenname, website, straßenr, ort, plz, land)`
	`Charge(cid, herstelldatum, hid#)`
	`Produkt(pid, bezeichnung, farbe, herstellerdatum, cid#)`
	`Eigenschaft(eid, einheit, bezeichnung)`
	`Pruefgeraet(gid, hersteller, bezeichnung, din, eid#)`
	`Mitarbeiter(mid, kuerzel, nachname, vorname)`
	`prueft(mid#, cid#)`
	`hat(eid#, pid#)`
	`benutzt(mid#, gid#)`
		4	1	

	begründen Alle Entitätstypen werden mit ihren Attributen in Relationen überführt. Für die 1:n-Beziehung von Prüfgerät und Eigenschaft (Prüfgeräte prüfen genau eine Eigenschaft, eine Eigenschaft kann von mehreren Prüfgeräten geprüft werden) wird der Primärschlüssel von Eigenschaft als Fremdschlüssel in die Tabelle Prüfgerät übernommen. Jede n:m-Beziehung zwischen zwei Entitätstypen wird in eine eigene Tabelle transformiert, wobei beide Primärschlüssel der Entitätstypen als Fremdschlüssel in die Beziehungstabelle eingehen und zusammen den Primärschlüssel bilden.		3	

Aufg.	**erwartete Leistungen**	**BE**		
		I	**II**	**III**
2.3	angeben In der Tabelle kommen Redundanzen u.a. bei den Mitarbeitern und der Probennummer vor. Dadurch kommt es zu Inkonsistenzen, d.h. es kann z.B. nicht fehlerfrei zugeordnet werden, ob die Probe 2203123 zum Hersteller HWP oder PWA Aschaffenburg gehört. beschreiben Eine Löschanomalie entsteht, wenn der Datensatz der Probe 2203123 gelöscht werden soll, da dabei auch die Informationen zum Hersteller verloren gehen. Eine Änderungsanomalie entsteht, wenn z.B. das Kürzel des Mitarbeiters von MA auf MO geändert werden soll. Sollte dabei ein Fehler unterlaufen führt dies zu Inkonsistenzen. Eine Einfügeanomalie entsteht, wenn nicht alle Daten vollständig vorliegen und deshalb nicht eingefügt werden können.	1	3	

2.4.1	formulieren
	`DELETE FROM eigenschaft`
	`WHERE einheit = 'Rho' AND bezeichnung = 'Massendichte';`
			3	

2.4.2	entwickeln
	`SELECT p.farbe, COUNT(p.pid) AS Anzahl`
	`FROM produkt p`
	`GROUP BY p.farbe`
	`HAVING COUNT(p.pid) > 5`
	`ORDER BY Anzahl DESC;`
			2	3

2.4.3	entwickeln
	`SELECT m.vorname, m.nachname, COUNT(p.mid) AS Anzahl`
	`FROM mitarbeiter m, prueft p`
	`WHERE m.mid = p.mid`
	`GROUP BY p.mid`
	`HAVING Anzahl > 500`
	`ORDER BY Anzahl DESC;`
			2	3

2.4.4	entwickeln
	`SELECT bezeichnung, hersteller`
	`FROM pruefgeraet`
	`WHERE gid NOT IN (`
	`  SELECT gid`
	`  FROM benutzt);`
			1	2

Nicht für den Prüfling bestimmt

Aufg.	erwartete Leistungen	BE		
		I	II	III
2.5	entwickeln, zeichnen			
	entwickeln		3	3
	zeichnen	1		
	Summe 40	**11**	**18**	**11**

III Bewertung und Beurteilung

Die Bewertung und Beurteilung erfolgt unter Beachtung der nachfolgenden Vorgaben nach § 33 der Oberstufen- und Abiturverordnung (OAVO) in der jeweils geltenden Fassung. Bei der Bewertung und Beurteilung der sprachlichen Richtigkeit in der deutschen Sprache sind die Bestimmungen des § 9 Abs. 12 Satz 3 OAVO in Verbindung mit Anlage 9b anzuwenden.

Bei der Bewertung und Beurteilung der Übersetzungsleistung in den Fächern Latein und Altgriechisch sind die Bestimmungen des § 9 Abs. 14 OAVO in Verbindung mit Anlage 9c anzuwenden.

Der Fehlerindex ist nach Anlage 9b zu § 9 Abs. 12 OAVO zu berechnen. Für die Ermittlung der Punkte nach Anlage 9a zu § 9 Abs. 12 OAVO sowie Anlage 9c zu § 9 Abs. 14 OAVO wird jeweils der ganzzahlige nicht gerundete Prozentsatz bzw. Fehlerindex zugrunde gelegt.

Für die Bewertung in den modernen Fremdsprachen ist der „Erlass zur Bewertung und Beurteilung von schriftlichen Arbeiten in allen Grund- und Leistungskursen der neu beginnenden und fortgeführten modernen Fremdsprachen in der gymnasialen Oberstufe, dem beruflichen Gymnasium, dem Abendgymnasium und dem Hessenkolleg“ vom 7. August 2020 (ABl. S. 519) zugrunde zu legen. Demnach erfolgt die Bewertung und Beurteilung mit der Maßgabe, dass lediglich bei der Ermittlung des Prüfungsergebnisses (Note) aus Prüfungsteil 1 und 2 gerundet wird.

Darüber hinaus sind die Vorgaben der Erlasse „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen (Abiturerlass)“, „Hinweise zur Vorbereitung auf die schriftlichen Abiturprüfungen im beruflichen Gymnasium (fachrichtungs-/ schwerpunktbezogene Fächer) (Abiturerlass BG)“ und „Durchführungsbestimmungen zum Landesabitur“ in der für den Abiturjahrgang geltenden Fassung zu beachten.

Als Kriterien für die Bewertung und Beurteilung dienen unter Beachtung der Zielsetzung der gymnasialen Oberstufe nach § 1 Abs. 2 OAVO neben dem Inhaltlichen auch die in den Kerncurricula genannten überfachlichen Kompetenzen, insbesondere die Sprachkompetenz und Wissenschaftspropädeutik; dies zeigt sich u.a. in qualitativen Merkmalen wie Strukturierung, Differenziertheit, (fach-)sprachlicher Gestaltung und Schlüssigkeit der Argumentation.

Im Fach Praktische Informatik besteht die Prüfungsleistung aus der Bearbeitung eines Vorschlags, wofür insgesamt maximal 100 BE vergeben werden können. Ein Prüfungsergebnis von **5 Punkten (ausreichend)** setzt voraus, dass mindestens 45% der zu vergebenden BE erreicht werden. Ein Prüfungsergebnis von **11 Punkten (gut)** setzt voraus, dass mindestens 75% der zu vergebenden BE erreicht werden.

Gewichtung der Aufgaben und Zuordnung der Bewertungseinheiten zu den Anforderungsbereichen für die Interpretationsaufgabe

Aufgabe	**Bewertungseinheiten in den Anforderungsbereichen**			**Summe**
	AFB I	**AFB II**	**AFB III**	
1	19	24	17	**60**
2	11	18	11	**40**
Summe	**30**	**42**	**28**	**100**

Die auf die Anforderungsbereiche verteilten Bewertungseinheiten innerhalb der Aufgaben sind als Richtwerte zu verstehen.